IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 59, No. 4, April, 2026, pp. 855-869
Journal homepage: http://iieta.org/journals/jesa

An Intrusion Detection and Defence Mechanism for Securing Autonomous Vehicle CAN Bus Using AI-Based Homomorphic Encryption

K. P. Senthilkumar*, E. Anbalagan

Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, SIMATS, Saveetha University, Chennai 602105, India

Corresponding Author Email: kpsenthilkumar05111983@gmail.com

https://doi.org/10.18280/jesa.590401

Received: 5 February 2026
Revised: 3 April 2026
Accepted: 14 April 2026
Available online: 30 April 2026

Keywords:
Controller Area Network, Batch Normalization, random drop imputation, Heat-diffusion for Affinity-based Trajectory Embedding, conservative physics-informed neural network, modified TrustNet, homomorphic encryption

ABSTRACT

Rapid technological advancements have transformed the automobile sector, shifting control from mechanical to software systems. Autonomous vehicles use the Controller Area Network (CAN) bus protocol for communication. The complexity of data and traffic makes these networks vulnerable to cyberattacks. Existing Intrusion Detection Systems (IDS) struggle with limited resources, slow response times, and low detection accuracy. Additionally, the growing number of Electronic Control Units (ECUs) increases security risks within in-vehicle networks. To address this problem, a homomorphic encryption-based machine learning technique can secure and detect attacks on in-vehicle CAN buses. CAN bus datasets are used as input data. The collected data is pre-processed using Batch Normalization (BN) and random drop imputation to improve quality. Potential of Heat-diffusion for Affinity-based Trajectory Embedding (PHATE) is applied for dimensionality reduction. The reduced data are fed into the Fully Homomorphic Encryption (FHE) named as Cheon, Kim, Kim, and Song (CKKS) based conservative physics-informed neural network (cPINN) algorithm to secure the system against cyberattacks in an autonomous vehicle. The modified TrustNet ensures low maintenance and defense against attacks. This classifier reached 97.37% accuracy and 98.02% specificity. Encryption and decryption times were 106.57sec and 47.28sec. This proposed approach is the best choice to secure and detect the IDS on vehicle networks.

1. INTRODUCTION

Modern technical developments in the automotive industry have focused on modern automobiles [1]. Because of this, the modern complexity of vehicles is rising and providing a greater range of cutting-edge, practical apps that encompass several functions. These characteristics are managed by hundreds of Electronic Control Units (ECUs), which are connected to one another over a Controller Area Network (CAN) bus [2]. Vehicle subsystems are monitored and controlled by the ECU to increase energy efficiency and lessen noise and vibration [3]. In order to safeguard communication from online threats, CAN lacks security features like authentication and encryption. Researchers have shown that in-car networks are seriously vulnerable to security flaws. An attacker can physically access a car through a variety of means, such as manipulating and accessing an Electronic Control Unit (ECU) through faulty designs or sending fake messages into the CAN system [4].

High connectivity and automobile electronics are developing as creative solutions to improve driving convenience [5]. Smart devices and cellular networks are used in vehicle-to-vehicle communication to transmit vital information, especially unsafe road conditions. Sensors in autonomous cars enable vehicle-to-infrastructure communication. Still, the improvements in automobile connectivity are susceptible to outside attacks, such as the absence of authenticating mechanisms in the current CAN messaging frame and the increasing complexity of in-car controller designs [6]. This might cause unwanted movements or failures, threatening the safety of passengers and vehicle cybersecurity.

Cybersecurity in essential contexts, such as automobiles, necessitates high-accuracy intrusion detection systems (IDSs) to avoid false alarms and safeguard passengers, pedestrians, and other drivers from malicious assaults [7]. Real-time reaction is critical for vehicle cybersecurity, but due to time and space limits, in-vehicle systems might not react in an instant. A real-time IDS is required to perform with restricted resources. The CAN bus technology contains technological flaws, rendering it susceptible to network-based assaults [8]. CAN systems have been developed by a number of businesses, including Google, Tesla, the University of Michigan, BMW, Audi, Mercedes Benz, and Baidu. Spoofing and flood attacks, which fall under the categories of passive and active attacks [9].

Physical aspects of CAN protocol security, such as access restrictions and encryption, have received the most attention

[10]. However, there is a need for more efficient IDS to address the limitations of CAN bus communications. To identify threats, traditional ML-based IDS algorithms such as Naive Bayes (NB), support vector machines (SVM), decision trees (DT), Multi-Layer Perceptron (MLP), and random forests (RF) are employed. Due to the limited processing capability of traditional ECUs, these algorithms are especially effective in vehicle networks. A systematic framework for detecting and classifying cyberattacks is still inadequate. However, employing a single base classifier, high-degree-of-freedom models may fail to match data distributions effectively. The computational cost of these options makes them unsuitable for use in critical systems like AV system. This proposed method presents a novel IDS for autonomous vehicles by integrating Fully Homomorphic Encryption (FHE) with a conservative physics-informed neural network (cPINN). Unlike traditional IDS approaches, which often require access to unencrypted data and may compromise privacy, the proposed system enables secure and accurate intrusion detection directly on encrypted CAN bus data. By leveraging the strengths of FHE and cPINNs, the system ensures data privacy without sacrificing detection accuracy or computational efficiency. This makes it particularly well-suited for resource-constrained environments typical of autonomous vehicle systems, providing enhanced cybersecurity for in-vehicle CAN bus networks.

Main contributions of the proposed technique are listed as follows:

- Homomorphic encryption-based machine learning is utilized to improve data security in Connected and Autonomous Vehicles (CAVs) by recognizing possible attack data.
- Random drop imputation and Batch Normalization (BN) are utilized as pre-processing techniques to impute the missing values and to standardize the given data.
- The PHATE technique is developed to reduce the dataset dimensionality.
- Homomorphic encryption, named as Cheon, Kim, Kim, and Song (CKKS) based cPINN is utilized to secure the information by identifying the possible attacks in CAVs.
- Modified TrustNet is utilized to mitigate attacks on CAN bus critical communications, ensuring minimal maintenance and protection from potential threats.
- For analyzing the efficiency of the proposed FHE-cPINN model, the performance of accuracy, precision, F1 score are significantly higher than existing approaches.

This paper is structured as follows: Section 2 provides a comprehensive summary of relevant work, including its shortcomings. Section 3 provides a brief overview of the proposed technique and its associated architectures. Section 4 contains the results and a discussion. Section 5 presents the conclusion and future scope.

2. LITERATURE REVIEW

This section provides an overview of the literature on anomaly detection approaches for protecting autonomous vehicles. Below are some of the most existing research and publications.

Altalbe [11] introduced a low-complexity intrusion detection solution for in-car networks called Feature Fusion and Stacking-based IDS (FFSIDS). Feature fusion and ensemble learning were used by FFSIDS to categorize traffic into invasive and non-intrusive groups. It uses a random forest as a metalearner and a decision tree as a basic classifier. The efficacy of FFSIDS in identifying intrusions was essential for guaranteeing the cybersecurity and security of contemporary automobiles. Attacks can be accurately detected with up to 95.5% accuracy. However, these methods use massive computing power and may fail to deliver excellent precision in network traffic categorization.

Olufowobi et al. [12] developed SAIDuCANT, a specification-based IDS that employs anomaly-based supervised learning, and identified a novel technique for extracting real-time model parameters of CAN buses. An open-source dataset of actual occurrences and authentic CAN recordings from two passenger cars were used to test the system. The results show that SAIDuCANT can identify data injection attacks 90% of the time with negligible False Positive Rate (FPR). But because of their enormous complexity, they were computationally costly.

Barletta et al. [13] provided a successful IDS that uses an unsupervised Kohonen Self-Organizing Map (SOM) network to identify attack messages on a CAN bus. The SOM network was appropriate for a wide range of applications due to its high detection rate, quick training time, and adaptability. In order to increase model accuracy, hybrid techniques like the k-means algorithm were applied and the system was expanded to include in-vehicle CAN buses. An automobile hacking dataset with a high traffic volume and an unbalanced data distribution was used to test the models. The developed method significantly enhanced network intrusion accuracy of spoofing the RPM gauge as 95.75% and reduced clustering iterations compared to the SOM network. Because of its great adaptability, low training time, and high detection rate, SOM network was often employed in security challenges.

Pascale et al. [14] presented a two-step algorithm-based integrated IDS for the automotive industry that can detect potential hacks. Using location and temporal analysis, the first step filters communications on the CAN bus. If malicious communications were identified, a Bayesian network analyzes them to determine the chance of a certain occurrence being categorized as an attack. The system performance has accuracy of 94.5%. However, the needed computational capacity exceeds the capabilities.

Song et al. [15] introduced a deep convolutional neural network (DCNN) based IDS designed to protect a vehicle's CAN bus. Without requiring human-designed characteristics, the DCNN identifies malicious traffic by learning patterns in network traffic. The Inception-ResNet model architecture becomes simpler while achieving good detection performance because of the model's optimization for CAN bus data flow. According to DoS, gear, and RPM attack datasets, the suggested IDS has a False Negative Rate (FNR) and ER of less than 0.1%; in the fuzzy dataset, it was 0.24%. But a model gets overly complex, uses too much processing power, and runs the danger of overfitting.

Jeong et al. [16] provided a method for detecting intrusions in automobile Ethernet networks using AVTP stream injection techniques. It was the initial time that such a mechanism has been created for automobile Ethernet. The model employs feature creation and a convolutional neural network. Real AVTP packets and a BroadR-Reach-based testbed were used to assess the system. The model was tested and produced the

best accuracy rates of 95.95%. However, Ethernet's transmission was insufficient for CAVs due to probable delays or traffic loss, which might be harmful in automotive applications. Table 1 shows the Overall comparison of the literature review.

Table 1. Overall comparison of the literature review

Author	Techniques	Performance	Drawbacks
Altalbe [11]	Feature Fusion and Stacking-based IDS (FFS-IDS)	Accuracy: 95.5%	FFSIDS systems demand tremendous computing resources.
Olufowobi et al. [12]	Specification-based Automotive Intrusion Detection using Controller Area Network (CAN) Timing (SAIDuCANT)	Accuracy: 87.8% Precision: 86.3% F1 score: 92.5%	Computationally expensive.
Barletta et al. [13]	Unsupervised Kohonen Self-Organizing Map (SOM)	Accuracy: 95.7% Precision: 80.41%	Security issues and minimal training time.
Pascale et al. [14]	Intrusion Detection System (IDS) for the automotive sector	Precision: 94.5% Recall: 93.4%	Processing power required by Bayesian networks exceeds their capabilities.
Song et al. [15]	Deep Convolutional Neural Network (DCNN)		Large amount of processing power and runs the risk of adjusting.
Jeong et al. [16]	Convolutional Neural Network (CNN)	Accuracy: 95.95%	Possible delays or traffic loss.

According to the papers described above, there are multiple significant challenges for detecting attack in autonomous vehicles. However, FFSIDS systems demand tremendous computing resources [11], Because of its enormous complexity, SAIDuCANT is computationally expensive [12], SOM networks are commonly used in security issues due to their versatility, and minimal training time [13], Processing power required by Bayesian networks exceeds their capabilities [14], DCNN method was extremely complex, demands a large amount of processing power and runs the risk of adjusting [15] and Ethernet transmission of CAVs might be insufficient because of possible delays or traffic loss [16]. In order to overcome these concerns, this proposed methodology leverages FHE through the CKKS scheme integrated with a cPINN. This approach aims to provide robust, accurate, and privacy-preserving intrusion detection on CAN bus networks, balancing high performance with practical resource management for enhanced vehicle cybersecurity.

3. PROPOSED METHODOLOGY

Modern vehicles are built with ECUs and sophisticated computing systems that enhance communication and comfort. However, greater connection raises security problems since the CAN is vulnerable to hackers. IDSs are crucial for protecting in-vehicle networks. Effective IDSs have been developed using machine learning and deep learning techniques. However, existing Machine Learning (ML)-based IDSs face issues such as poor detection accuracy, delayed real-time response, and inadequate processing power due to network traffic quirks. Therefore, encryption-based supervised ML classification algorithms are essential in solving the challenges with data security in CAVs by determining anticipated messages of attack. The proposed approach for securing and detecting threats against a CAV network is depicted in Figure 1, and its working flow is given in Figure 2. Initially, CAN Bus dataset is collected from real CAN traffic data. Random drop imputation, and BN is utilized as pre-processing techniques to impute the missing values and standardize the input data. Then, dimensionality of the dataset gets reduced by PHATE. These reduced data are trained using homomorphic encryption named as CKKS based on cPINN that secures and diagnose attacks on CAN buses in vehicles. A proposed prediction model can detect whether there is attack or non-attack. Once an attack has been detected, the modified TrustNet is employed to mitigate the attack. Modified TrustNet attempts to protect CAN-bus critical communications from attacks with minimal maintenance. The diagram presents a comprehensive pipeline for detecting attacks in a CAN bus dataset. The process begins with preprocessing, where missing values are addressed using Random Drop Imputation and the data is standardized through BN to ensure consistent scaling across features. Following preprocessing, the high-dimensional dataset undergoes dimensionality reduction using PHATE (Potential of Heat-diffusion for Affinity-based Transition Embedding), which effectively reduces the feature vector while preserving essential data structure. The reduced data is then fed into a classification module that combines homomorphic encryption with a cPINN. This enables secure and privacy-preserving classification of the data into either "attack" or "non-attack" categories. Detected attacks are subsequently handled by a Modified TrustNet framework, which likely further evaluates or mitigates the identified threats, ensuring secure and trustworthy decision-making within the system.

3.1 Pre-processing

A crucial step in the data mining process is data pre-processing. Before data is analyzed, it must first be combined, cleaned, and converted. Increasing the data's quality and applicability for the particular data mining activity is the goal of data preparation. In this proposed approach, Random drop imputation (RDI) is used to replace missing values in a raw input dataset, and the data is adjusted within a certain range using BN.

3.1.1 Random drop imputation

An imputation model is explicitly trained by RDI using random drop data, which is produced by arbitrarily eliminating observed values from the time-series data [17]. Let's construct $\tilde{X} = \{\tilde{x}_1, \ldots, \tilde{x}_T\} \epsilon \mathbb{R}^{T \times D}$ to represent the random drop data, where each element is a vector $\tilde{x}_t = \{\tilde{x}_t^1, \ldots, \tilde{x}_t^D\} \epsilon \mathbb{R}^D$. Additionally, designate $\widetilde{M}$, a mask whose element $\widetilde{m}_t^d$ is specified as follows, and which indicates the position of both the initially missing values and the randomly dropped data:

$$\widetilde{m}_t^d = \begin{cases} 0, & if\ \tilde{x}_t^d\ as\ missing\ value \\ 1 & otherwise \end{cases} \quad (1)$$

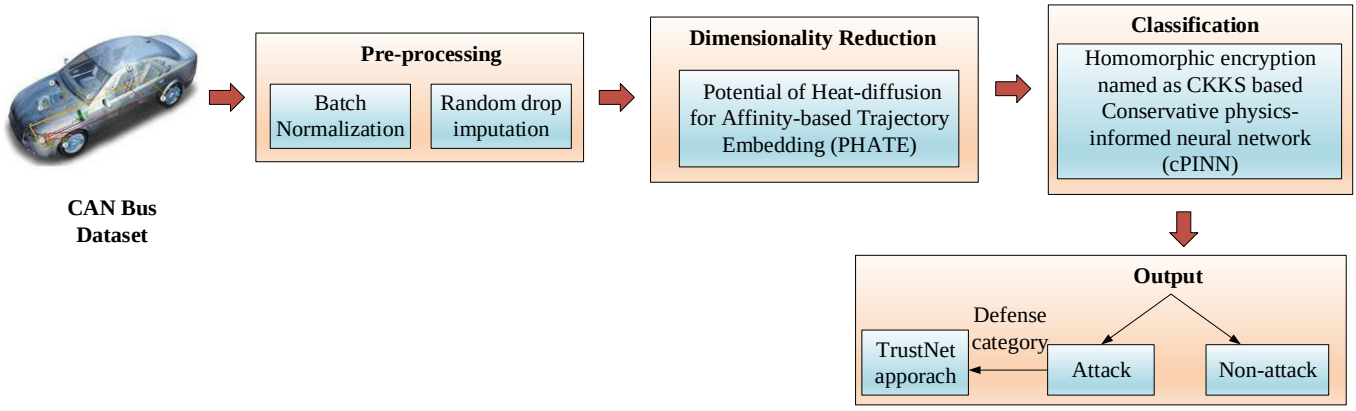

Figure 1. Design of the proposed ML-based IDS to secure and detect attacks against a CAV Network
Note: ML = Machine Learning; IDS = Intrusion Detection System; CAV = Connected and Autonomous Vehicle

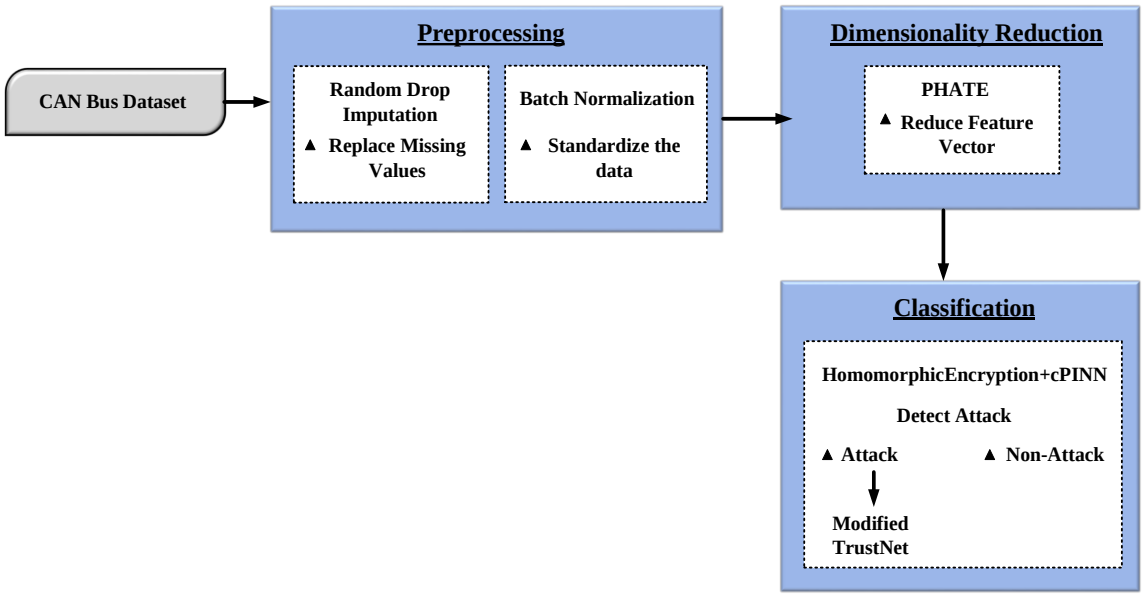

Figure 2. Working flow of the proposed method

The imputation loss, also known as the loss function of RDI, has the following expression:

$$L_{impute}(\tilde{X}, \tilde{M}) = \left\| X \odot (M - \tilde{M}) - F(\widetilde{X;\theta}) \odot (M - \tilde{M}) \right\|_2 + \left\| \tilde{X} \odot \tilde{M} - F(\widetilde{X;\theta}) \odot \tilde{M} \right\|_2 \quad (2)$$

where, θ is the parameter of an imputation model F. By producing missing values with ground truth, the proposed technique, RDI, enables the explicit training of an imputation model. Additionally, it provides data augmentation by creating random drop data from the source data. Through ensemble learning, the supplemented data is used. Although it has bias difficulties, bootstrap with ensemble learning was employed to overcome unstable and over-fitting issues. However, RDI augments the whole dataset to produce distinct sets, making it possible to apply ensemble learning without bias. The procedure is as follows: N imputation models F_k are trained with various data using the loss function (2), and N random drop data, represented as $\tilde{X}_k$, are generated. The drop data is used to create original data for each model F_k, which is then provided to each model Fk that has already been trained for imputation. Since each model's output values are unique, the ensemble model's ultimate output is determined by averaging its N output values.

3.1.2 Batch Normalization

BN is a technique for minimizing the internal covariate shift induced by changes in input signal distribution [18]. Instead of utilizing the statistics of the complete dataset to normalize intermediate representations, a mini-batch is used to lower processing costs. The mini-batch's input characteristics are normalized using the determined statistics as,

$$\hat{x}_b = \frac{x_b - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (3)$$

where, ϵ is a numerical stability-related tiny constant. The input characteristics, which can be expressed as $BN(x_I) = \gamma \hat{x}_i + \beta$, are transformed using the learnable parameters γ *and* β to further enhance the layer's representation capacity. BN uses the running average of the training mean and variance during the inference stage. In contrast to the static BN, this study uses SNNs to enable temporally-varying parameters in BN, allowing exploration of the temporal properties of BN.

3.2 Dimensionality reduction

Dimensionality reduction is a strategy for reducing the number of features in a dataset while maintaining its original qualities. To develop a model with fewer variables,

unnecessary or redundant characteristics, as well as noisy data, are removed. This technique includes a variety of feature selection and data compression strategies employed during pre-processing, all of which reduce high-dimensional spaces to low-dimensional ones. In this work, an approach known as PHATE decreases dimensionality and visualizes data while retaining both local and global data structures.

3.2.1 PHATE

A novel feature analysis viewpoint for deep forest classification is provided by the dimensionality reduction technique known as PHATE, which maintains local relationships between data points while learning general spatial features. In line with other techniques, it reduces the eigenvector dimension to 32 [19]. The nonlinear, unsupervised method known as PHATE maintains local and global data links and accurately represents high-dimensional data sets while combining the advantages of PCA and tsne.

The following are the particular actions.

An expression for the eigenvector of a series is $x_n\, n, \in 1, \ldots, k$, where k = 200. The Gaussian kernel function quantifies the similarity between x_a and x_b, $a, b \in \{1, \ldots, k\}$, based on the Euclidean distance between them, expressed as $k_z\,(x_a, x_b)$.

$$k_z(x_a, x_b) = \exp\left(-\frac{\|x_a - x_b\|^2}{\epsilon}\right) \tag{4}$$

wherein the neighborhood radius that the kernel function captures is determined by the bandwidth measurement, denoted by ε.

Reduce stress by using metric multidimensional scaling (MDS) measurements $(\hat{x}_1, \ldots, \hat{x}_{32})$,

$$stress(\hat{x}_1, \ldots, \hat{x}_{32}) = \sqrt{\sum_{i,j}(\Re^t_{(x_i,x_j)} - \left\|\hat{x}_i - \hat{x}_j\right\|_J)^2 / \sum_{i,j}(\Re^t_{(x_i,x_j)})^2} \tag{5}$$

A 32-length fixed-length vector has been generated by capturing the data in the MDS embedding. In order to identify message attacks in CAN and secure the vehicle's network from cyber threats, these reduced dimensionality data are trained using homomorphic encryption, named as CKKS based cPINN.

3.3 Classification

Reduced data serves as input for the classification process, which secures and predicts if the network is attacked or not. A classifier is an algorithm that sorts data into labeled groups or categories of information. The proposed methodology uses homomorphic encryption named as CKKS based cPINN, which has undergone extensive training and evaluation on test and training data.

3.3.1 Homomorphic encryption

For real numbers, a previously explained homomorphic computing algorithm is homomorphic encryption for Arithmetic of Approximate Numbers (HEAAN), sometimes referred to as CKKS [20]. Ring-Learning with Errors (Ring-LWE) employs noise, sometimes known as error e, to rely on FHE methods. This noise is encrypted as part of the payload for security purposes. The payload and noise are combined to generate the plaintext (μ + e), which is then encrypted.

In CKKS, the space between the plaintext and the ciphertext is nearly equal. These components are based on the polynomial ring $R_S = \mathbb{Z}_S\,(y)/f(y)$, where s is the integer coefficient modulus and $f(y)$ is exponent of polynomials. The polynomials that make up R_S have integer coefficients that are constrained by s. In addition, the degree of $f(y)$ limits their degrees. With k being a power of two known as the ring dimension, the formula for $f(y)$ that is most frequently seen in research is $f(y) = y^k + 1$. The number of ring components changes with each appearance of the CKKS ciphertext and plaintext. The CKKS plaintext has a single ring element (polynomial), while the CKKS ciphertext contains two or more ring elements.

CKKS provides encoding and decoding (codec) algorithms that allow a vector of real numbers, or more precisely, complex numbers, to be translated between plaintext space and another vector. Integral operands can be used in combination with integer operations to carry out fixed-point arithmetic operations, which are represented by CKKS.

1. Plain Text Encoding and Decoding

The CKKS investigation is further enhanced by a novel codec technique that converts a vector of complex numbers into an equivalent plaintext item and vice versa. One single plaintext element, $b \epsilon R$, is produced during the encoding of $a\frac{k}{2}$ vector containing complex numbers, $y \in c^{\frac{k}{2}}$.

Decoding was the process of reversing a component of plaintext and yielding a series with complicated numbers. Procedure makes use of Eq. (12). A version of the Fourier transform is the complex canonical embedding, or map π.

$$encode(y, \Delta) = \lfloor \Delta . \pi^{-1}(y) \rceil \tag{6}$$

$$decode(b, \Delta) = \pi\left(\frac{1}{\Delta} . b\right) \tag{7}$$

The relationship between the fixed-point format and the CKKS codec method ought to be clear. In order to decrease the smallest fractional components during encoding, rounding is performed. Scaling is accomplished by multiplying the supplied payload by Δ. In decoding, it's a reverse method. When encoding, multiply by Δ, and when decoding, divide by Δ.

2. Parameters

Three variables were analyzed: k, s, and Δ. R_2 is used for evenly sampling polynomials $\{-1, 0, 1\}$ with coefficients of integers in CKKS, whereas 𝕏 is used for a distribution that is gaussian discontinuity across R utilizing variables $\mu\ and\ \sigma$, constrained by an integer β.

It has been determined that the present homomorphic encryption norm [ACC+18] as (μ, σ, β) is $\left(0, \frac{8}{\sqrt{2\pi}} \approx 3.2, [6 \cdot \sigma] = 19\right)$.

R_s represents an attire randomized over. When q is fixed, the study finds that an RLWE hardness estimate with n offers adequate security. A ring's coefficient modulus is expressed as $s' = \frac{s}{\Delta}$ and its size is decreased as a result of the homomorphic multiplication process, which reduces the ciphertext by Δ.

A connection among ciphertext coefficients, where $1 \leq v \leq V$, and V denotes a recently encoded cipher text level, is the coefficient modulus at level v, shown by s_v.

$$s_V > s_{V-1} > \cdots > s_1 \tag{8}$$

3. Key Generation

An interval-based quadratic with degree multiple variables $\{-1, 0, +1\}$ is used to construct the secret key S_k. Communication encryption is done using AES 256 bit encryption with randomness and predefined settings. Each of the 14 rounds that comprise the AES-256 encryption key generates a 128-bit round key. (P_{k1}, P_{k2}) is a set of polynomials that represents the public key PK that was derived using the given parameters.

$$P_{k1} = [-b.S_k + e]_{sV} \tag{9}$$

$$P_{k2} = b \leftarrow^{g} \quad R_{sV} \tag{10}$$

Polynomial computation should be performed modulo sV, or more specifically modulo the ring polynomial modulus $y^k + 1$, since P_{k2} is in R_{sV} and e is an unpredictability polynomial that $\mathbb{X}$ chose. This is evident from the symbol $[.]_{sV}$.

4. Encryption and Decryption

Either R (which encrypts the input payload vector) or plaintext message m can be encrypted. To get the ciphertext $c = (c_1, c_2)$, three shorter randomized polynomials (g from R_2 and e_1 and e_2 from $\mathbb{X}$) should be constructed in R^2_{sV} in the sense described below:

The secret key is evaluated to estimate the length of the message in plaintext during decryption of the provided ciphertext at stage v.

$$\hat{m} = [c_1 + c_2 \cdot S_k]_{sV} \tag{11}$$

5. Evaluation of Homomorphic

Addition and multiplication using homomorphism are two kinds of homomorphic operations that are carried out.

a) Eval-Add

Eq. (20) illustrates that input cipher messages just need to be treated to the necessary polynomials. This should be taken into account because the input cipher texts are probably equivalent in terms of level and scale. If not, the ciphertext levels at higher levels must be decreased to match those at lower levels.

$$eval - add\left(c^{(1)}, c^{(2)}\right) = ([c_1^1 + c_1^2]_{sV}, [c_2^1 + c_2^2]_{sV}) = (c_1^3, c_2^3) = c^3 \tag{12}$$

6. Eval-Mult

Eval-Mult may be calculated using two cipher messages and the following technique:

$$eval - mult\left(c^{(1)}, c^{(2)}\right) = ([c_1^1.c_1^2]_{sV}, [c_1^1.c_2^2 + c_2^1.c_1^2]_{sV}, [c_2^1.c_2^2]_{sV}) = (c_1^3, c_2^3, c_3^3) = c^3 \tag{13}$$

Relinearization uses R_{sV} for calculation to minimize the size of c^3 polynomials in a ciphertext. The product is encrypted using the non-expanded ciphertext with a squared scale factor of Δ^2. To repeat the operation, two ciphertexts in CKKS can be multiplied with various $\Delta_1 \cdot \Delta_2$ scale factors. It is necessary to make sure the product does not wrap around or fit inside the datatype in order for this fixed-point arithmetic approach to be practical. It is suggested that for easy representation, a fixed scale factor be used throughout the calculation.

3.3.2 Conservative physics-informed neural network

An expansion of PINN designed to solve the conservation lw is called cPINN. It addresses the problems of PINN's accuracy and expensive training costs. Domains are subdivided into non-overlapping subdomains that take into account various neural network topologies. This makes it possible to choose networks for each subdomain more effectively. After that, the solution in each subdomain is reassembled with the appropriate interface conditions. Because of its unique purpose, which permits different system and scalar architectures in each subdomain, cPINN is still utilized for controlled equations even though it hasn't been studied extensively on hyperbolic conservation laws [21].

Let $N^L: \mathbb{R}^{Di} \rightarrow \mathbb{R}^{Da}$ be a neural network based on feed forwarding with L layers and N_k neurons in the kth layer$(N_0 = D_i, and\ N_L = D_a)$. The kth layer $(1 < k \leq L)$ weight matrix and bias vector are represented, respectively, by $W^k \epsilon \mathbb{R}^{N_k \times N_{k-1}}$ and $b^k \epsilon \mathbb{R}^{N_k}$. Denoted by $z \in R^{Di}$ for the input vector and $N^k(z)$ and $N^0(z) = z$ for the output vector at the k[th] layer, respectively. The scaling factor (n) and the scalable parameters na^k are applied layer-wise, with Φ representing the activation function. Additional parameters a^k change activation function slope in hidden-layers, increasing training speed and contributing to loss function through slope recovery term.

$$S(a^k) = \frac{1}{\frac{1}{L}\sum_{k=1}^{L} N(a^k)} \tag{14}$$

where, the operator for the exponential is N. The network's ability to learn is improved by such locally adaptable activation functions, particularly in the early training phase. A mathematical instance of this may be provided by contrasting the slope behavior of the static activating technique with the adjustable function of the activation approach. The gradient-dependent dynamics of adaptable activation differ from those of fixed activating by dividing an induced vector by a slope and introducing the predicted second-degree component. The present study, initialized $na^k = 1, \forall k$, and employed scaling factor $n = 5$ for every hidden layer. Figure 3 displays the cPINN algorithm design.

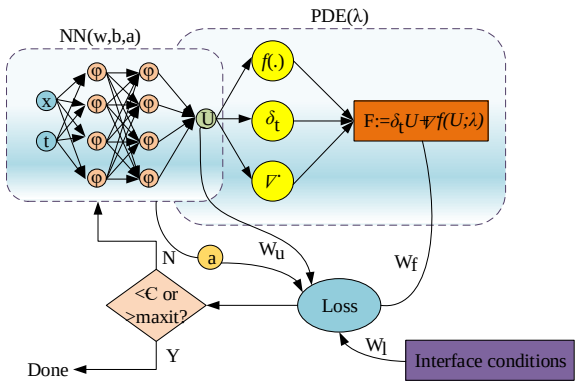

Figure 3. Adaptive activation function-based conservative physics-informed neural network (cPINN) schematic

The groundbreaking technology known as cPINN separates the domain into small sub-domains and enables the use of many neural networks with different topologies to solve PDEs. This method greatly accelerates convergence rates and increases computing efficiency. By pulling them from zero-

mean, finite-variance transportation, biases and weights are initialized using the Xavier initialization technique. For each given layer with N nodes, the optimal variance value is 1/N. Methodology also allows for the selection of network hyper-parameters depending on solution regularity, such as optimization method, activation function, depth, or width. This enables the deployment of shallow networks for smooth zones and deep networks for complicated areas.

The purpose of neural network training is to make use of available training data, which can be gained by starting and boundary conditions, properly executed experiments, or high-resolution numerical experiments. The data set is classified as low or high-fidelity depending on measurement inaccuracy and can be utilized to solve PDEs.

Sub-domain optimization technique and loss function:

The cPINN algorithm learns a substitute, $u = u_{\widetilde{\circledast}}$, for predicting the solution u for any input x_u^i given a PDE model and a set of training data, $\{u(x_u^i)\}_{i=1}^M, x^i \coloneqq (x_u^i, y_u^i, t_u^i) \sim p(x)$. Typically, one must approximate the density $p(x)$ using the input training data as it is not known a priori in most circumstances. The loss function, which is equipped with interface conditions and resembles the PINN loss in structure, may be specified subdomain-wise. Each sub-domain's loss function for the forward problem is provided by

$$\mathcal{L}(\widetilde{\circledast})_{pth\ sub-domain} = W_{u_p} \times MSE_{up} + W_{F_p} \times MSE_{F_p} + W_1 \times \underbrace{(MSE_1 + MSE_{u_{avg}})}_{Interface\ conditions} \quad (15)$$

where, W_{F_p}, W_1, and W_{u_p} denote the interface, residual, and boundary/initial weights, respectively. Using these classification approaches, attacks and non-attacks in CAN are detected, thereby securing the vehicle network from cyber threats. If an attack has been identified, it is defended by using the modified TrustNet.

3.4 Modified TrustNet

In Cyber-Physical Systems (CPS) networks based on CAN, Modified TrustNet is a secure technique that guards against cyberattacks. High safety standards are achieved without adding any additional information to the communication frames; instead, a conventional CAN bus with security features implemented in software or hardware is used. One way to maintain compliance with CPS's real-time requirements is to fully integrate Modified TrustNet in software [22]. Without requiring any hardware or architectural modifications, it provides minimal overhead protection against masquerade and replay attacks. Supporting all CAN frame IDs specified in the original CAN standard, modified TrustNet offers complete reversibility. Although CAN protocols are used in many applications, they were originally designed as an in-vehicle network. Malicious assaults on open systems that could lead to future issues or compromise the privacy of sensitive data must be addressed. Conventional security methods might not work due to CAN's physical and restricted nature and the requirement for real-time processing by inexpensive, low-performance microcontrollers. Below Figure 4 shows the legacy CAN peripheral is a component of middleware that communicates with TrustNet.

To enable modified TrustNet, which virtualizes the CAN bus and allows for both safe and insecure communication over the same physical channel, both filters and masking on outdated CAN devices need to be adjusted to 0. To maintain safe communication, modified TrustNet requires that reliable ECUs share private keys. Modified TrustNet segment nodes sync on a regular basis, according to the time intervals established by CAN frame transmissions on the CAN bus. Nodes keep broadcasting frames unless a synchronization frame indicates that a fresh period is in effect. The alignment frame is encoded as an unsecured CAN frame, as specified.

- ID:0 × 0
- DLC==0 × 1: Index of the Keys Table determined by the highest entry, duration set to 100 frames

LS portion's keys database index and the MS component's time interval make up the split payload for DLC > 0 × 1.

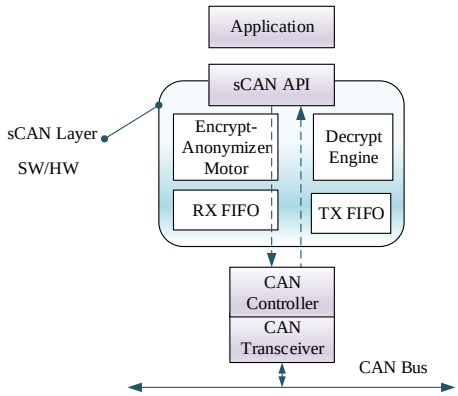

Figure 4. An Electronic Control Unit (ECU) that is based on an antiquated Controller Area Network (CAN) peripheral and uses technology to enable TrustNet

Arbitrators on the CAN bus favor cycles with dominating bits over negative ones, making it susceptible to attack. By ensuring the encryption process functions properly, using a CAN ID of 0 guards against malicious attacks. CAN controllers provide a separate Tx and Rx route, allowing applications to send and store CAN frames. The program manages the transmission process using "send-and-forget" functionality. Because Trust-Net encryption operates atomically in a single CAN structure, updating counters and HMAC data, synchronization issues might arise.

Separating the Tx and Rx procedures and using two separate counters for each direction is the next way to address the characteristics of CAN controllers. To do this, one counter must be added for each tuple of communication nodes and direction, such as (TxA, RxB) and (RxA, TxB). A node must keep as many counters per connection/direction if it is set up to communicate with many (or ALL) other nodes on the CAN bus. After the payload and CAN ID are extracted at reception, the CAN ID is subjected to an XOR operation using ALL valid counters at the RX route. By comparing the latest calculated CAN IDs with the list of valid IDs, XOR computations and search cost are minimized. This method effectively prevents replay attacks. Algorithm 1 provides the pseudo code for the proposed algorithm for securing and detecting attacks against a CAV network.

Algorithm 1: Pseudocode for securing and detecting attacks against a Connected and Autonomous Vehicle (CAV) network
Input: D = CAV data

# Pre-processing
{
RDI = Random drop imputation (D) // impute the missing values using Eqs. (1) to (2)
BN = Batch Normalization (RDI) // standardize the input values using Eqs. (3) to (5)
}
#Dimensionality reduction
{
PHATE = Potential of Heat-diffusion for Affinity-based Trajectory Embedding (BN) // Eqs. (6) to (11) can be used to reduce dimensionality.
}
#Classification
{
HEcP = homomorphic encryption named as CKKS based cPINN (PHATE) // Secure and detect the attacks against a CAV network.
}
#Defence Mechanism
{
MTN = Modified TrustNet (HEcP) // Defends against attacks
}
Output: Secure and detect attacks against a CAV network
End

4. RESULT AND DISCUSSION

Modern technologies are transforming sectors such as agriculture, health care, and industry, with autonomous vehicles (AVs) continually improving performance. AI-powered sensors monitor and supervise autonomous vehicle settings, identifying accidents and notifying drivers. However, public communication can cause security vulnerabilities such as DDoS, MITM, and cross-site scripting. CAN improve fault monitoring and detection, but they can also expose AVs to security attacks. To overcome these issues, the proposed system aims to secure the system against cyberattacks. Initially, the raw input data was preprocessed using random-drop imputation and BN. The dataset's dimensionality was reduced using PHATE, and the model was trained using Homomorphic Encryption (CKKS), named CKKS-based cPINN, to secure and detect attacks. A modified TrustNet approach is utilized to mitigate the attacks. Python 3.8 is used for testing, and the system specifications include an Intel Core i5 CPU, an NVIDIA GeForce GTX 1650 GPU, a 16-bit operating system, and 16GB of RAM.

i) Dataset

Data were obtained from the Car-Hacking Dataset for intrusion detection [23]. These datasets were produced by recording CAN traffic from an actual car and using regular packets, fuzzing, spoofing, and replaying as message insertion techniques. The datasets comprised 300 incursions, each lasting three to five seconds, and included CAN traffic, yielding a total of 30 to 40 minutes of data. This dataset has 3,00,000 data in total. Of this total, 80% (2,40,000) is used for training, while 20% (60,000) is used for testing.

cPINN classifier parameters are listed in Table 2. The hidden layer in the proposed techniques is 3, the learning rate is 0.01, and the classifier's optimizer is known as Adam. The output layer's activation function is softmax, the output layer's activation function is ReLu, and the loss is 'sparse_categorical_crossentropy'.

Table 2. Parameters of conservative physics-informed neural network (cPINN)

Hyper Parameter	Selection
Hidden layer	3
Learning rate	0.01
loss	'sparse_categorical_crossentropy'
Optimizer	Adam
Input layer's activation function	ReLu
Output layer's activation function	Softmax

4.1 Classification performance

The accuracy of a classification system can be assessed using a confusion matrix. Determining the regions where the classification strategy performs well or poorly may be made easier with the estimation of error matrices. The technique's class categorization yields both valid and incorrect classifications based on the provided data. Using confusion matrices, basic predictive metrics, including specificity, precision, accuracy, and recall, are displayed. The confusion matrix presented compares the predicted and actual labels. Figure 5 displays the confusion matrix for the proposed strategy. For the normal, fuzzy, and DDoS classes, the expected values are 19806, 19721, and 18109, respectively.

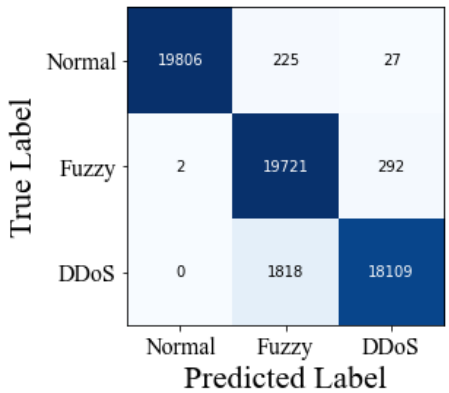

Figure 5. Confusion matrix for the proposed technique

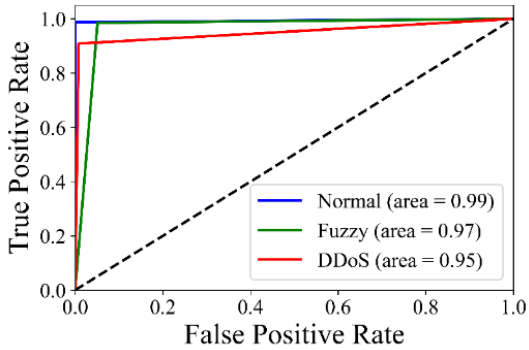

Figure 6. Proposed technique's Receiver Operating Characteristic (ROC) plot

The proposed approaches are shown in Figure 6, along with Receiver Operating Characteristic (ROC) and AUC curves. The validity and efficiency of a multi-class classification

problem are evaluated using the ROC and the area under the curve (AUC). AUC indicates the degree or quantity of separability. AUC around 0 indicates an excellent model. The proposed approach provides a good level of separability, as evidenced by the AUC plot, which is close to 1. The ROC curve shows that the model with normal, fuzzy, and DDoS traffic has comparable performance. Normal data occur at 99%, fuzzy at 97%, and DDoS at 95%.

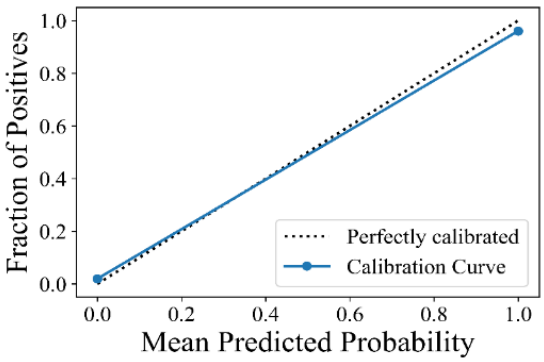

Figure 7. Calibration plots for the models

Figure 7 shows the calibration curve of the model's dependability. The curve is constructed using binned data, with the X-axis representing the average probability of the test dataset and the Y-axis representing the proportion of positives. If the model works properly, the curve will be a straight line with a gradient of 1.

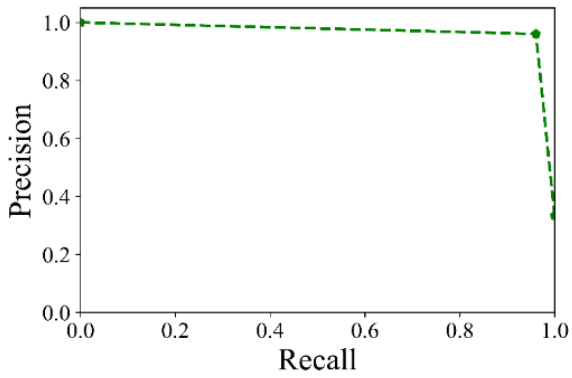

Figure 8. Precision-recall curve

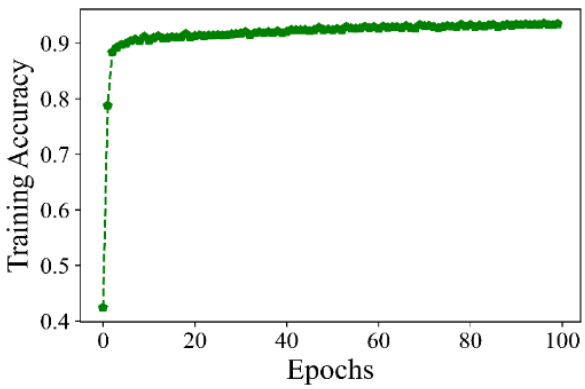

Figure 9. Training accuracy

Figure 8 presents the accuracy-recall for the developed technique. One way to see how a classifier's threshold selection affects its performance is to look at the precision-recall curve. It also shows how threshold selection affects classifier efficiency, making it simpler to decide which threshold is best in a specific situation.

Figure 9 depicts a training accuracy plot for the proposed approach. When the number of epochs is between 0 and 100, training accuracy remains constant. It increases from 41% to 98%, then remains steady. Similarly, Figure 10 depicts the training loss plot for the proposed technique. As the number of epochs increases, the training loss decreases and fluctuates. After some time, it remains steady.

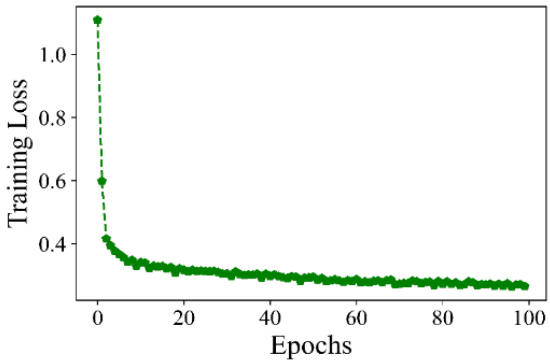

Figure 10. Training loss

4.2 Performance evaluation of the encryption and decryption algorithm

A proposed homomorphic encryption technique, CKKS, is used to secure data processing. This proposed CKKS algorithm has been evaluated using effectiveness measures, including encryption and decryption times. The proposed and existing encryption algorithms, such as the Hill cipher, Rivest-Shamir-Adleman (RSA), and the Paillier cryptosystem. Performance and results for the proposed and different existing encryption algorithms will be given following

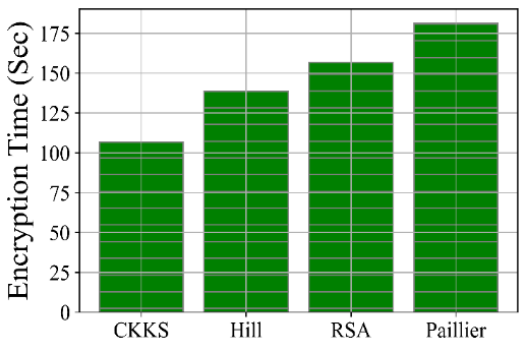

Figure 11. Comparison of encryption time

Figure 11 compares the encryption times of the proposed and existing methods for securing data. CKKS algorithms are now in use, and the existing algorithms include Hill, RSA, and Paillier. The proposed CKKS approach achieved an encryption time of 106.57 seconds. The previous encryption approaches required 138.55 seconds for Hill, 156.66 seconds for RSA, and 181.18 seconds for Paillier. The CKKS method requires less time than the existing algorithm. The CKKS algorithm achieves lower encryption time by operating on batched data using optimized polynomial operations in a ring structure.

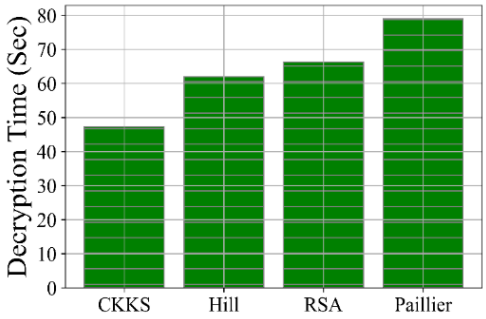

Figure 12. Comparison of decryption time

Figure 12 shows an evaluation of decryption times for proposed and existing techniques used for secure data transfer. To analyse decryption time (sec), the proposed CKKS was compared with existing methods, including Hill, RSA, and Paillier. The proposed CKKS approach needed 47.28 seconds to decipher. The proposed CKKS was then compared with the existing techniques for 61.94 sec, 2.883 sec, 66.19 sec, and 78.96 sec, including Hill, RSA, and Paillier. The proposed CKKS technique demands less time for decryption than existing algorithms. CKKS's design allows efficient decryption by leveraging homomorphic properties and polynomial approximations that avoid heavy modular arithmetic.

4.3 Comparison analysis for attack prediction

Several indicators were used to assess the efficacy of the proposed detection method. The measures used in this study are Net Present Value (NPV), F1-score, FPR, sensitivity, selectivity, accuracy, precision, specificity, error, MK, J-statistic, False Discovery Rate (FDR), F-Measure (FM), False Omission Rate (FOR), and false negative rate. The proposed homomorphic encryption approach based on cPINN (FHE-cPINN) is compared with several well-known methods, including DT, RF, k-nearest neighbors (KNN), SVM, and NB.

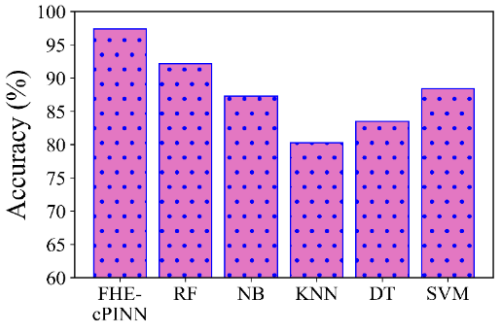

Figure 13. Analysis of accuracy

Figure 13 compares the accuracy metrics of the existing approach with the proposed FHE-cPINN. The accuracy of the proposed FHE-cPINN classifier is 97.37%. However, the accuracy ratings of the currently in use classifiers are 92.12%, 87.3%, 80.3%, 83.5%, and 88.4%, respectively, for RF, NB, KNN, DT, and SVM. It has been demonstrated that the proposed classifier achieves greater accuracy than the classifiers already in use. FHE-cPINN achieves higher accuracy by combining physics-informed constraints with encrypted neural networks, enabling better modeling of data distributions and reducing overfitting. Figure 14 displays the precision metrics for FHE-cPINN and all other approaches. The precision obtained with the FHE-cPINN technique was 96.29%. The existing classification designs, consisting of RF, NB, KNN, DT, and SVM, have precision values of 88.32%, 86.87%, 85.45%, 84.5%, and 86.1%, respectively. FHE-cPINN's improved precision reflects a lower false-positive rate, enabled by its use of domain knowledge via PINNs, resulting in more confident and selective predictions than traditional methods.

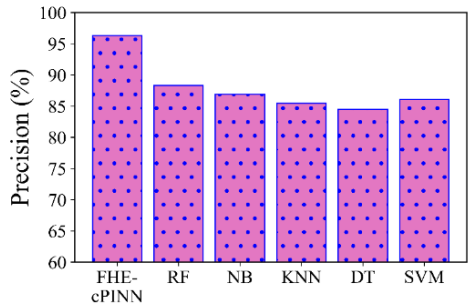

Figure 14. Analysis of precision

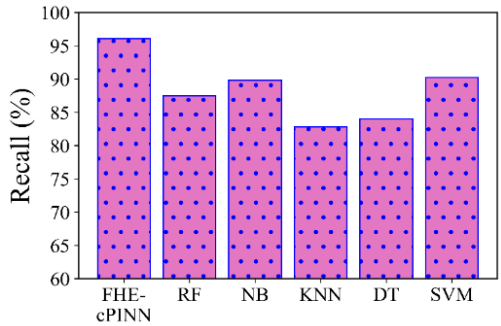

Figure 15. Analysis of recall

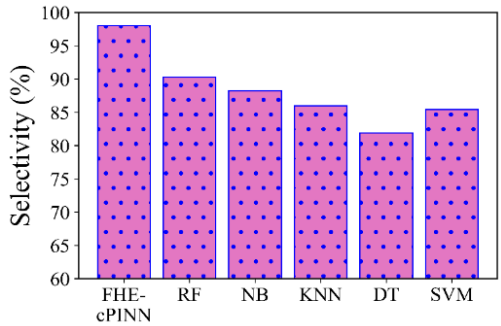

Figure 16. Analysis of selectivity

Figure 15 shows the recall metrics for the proposed and existing methods. The proposed method's FHE-cPINN classifier has a 96.05% recall rate. Using the existing method, the recall values of the RF, NB, KNN, DT, and SVM classifiers are 87.45%, 89.78%, 82.81%, 83.97%, and 90.23%. FHE-cPINN's higher recall demonstrates its ability to accurately detect true positives, driven by the enhanced representational power of physics-informed models and secure, encrypted learning. Figure 16 compares the proposed and existing selectivity results. With a selectivity value of 98.02%, the proposed FHE-cPINN classifier outperforms other commonly used methods, including RF, NB, KNN, DT, and SVM, which have specificity values of 90.3%, 88.24%,

85.98%, 81.9%, and 85.43%, respectively. This comparative analysis shows that the proposed FHE-cPINN classifier outperforms different methods. Selectivity measures the correct identification of negatives. FHE-cPINN's superior selectivity comes from using domain constraints to reduce false alarms, allowing it to more reliably distinguish benign events from attacks than traditional classifiers.

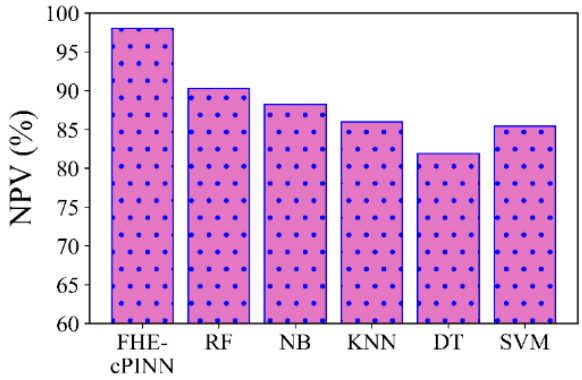

Figure 17. Comparison of Net Present Value (NPV)

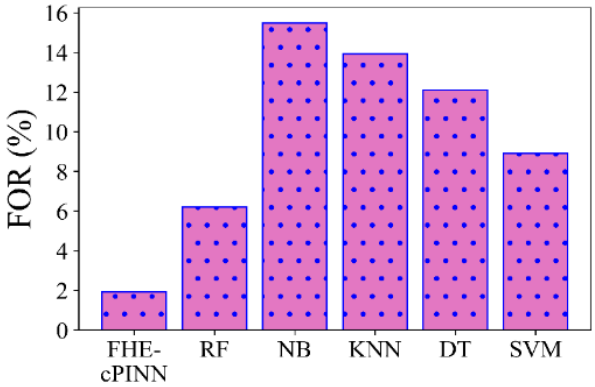

Figure 18. Comparison of False Omission Rate (FOR)

Figure 17 displays the NPV of the existing and FHE-cPINN approaches. The FHE-cPINN classifier used in the proposed approach has an NPV value of 98.07%. However, the NPV values for the currently in use RF, NB, KNN, DT and SVM algorithms are 93.8%, 84.52%, 86.08%, 87.9% and 91.1%, respectively. It implies that the FHE-cPINN strategy is more effective than existing strategies. A higher NPV for FHE-cPINN indicates greater confidence in predicting no-attack cases, thanks to the model's strict constraints and encrypted data privacy, thereby enhancing generalization to unseen benign samples. Figure 18 shows the FOR measurements of the comparison approaches. A yield of 1.93% was obtained with the proposed FHE-cPINN classifier. However, using these specific methods, the FOR values utilizing RF, NB, KNN, DT, and SVM are 6.2%, 15.48%, 13.92%, 12.1%, and 8.9%, respectively. Lower FOR in FHE-cPINN indicates fewer missed attacks, thanks to its improved ability to detect subtle malicious patterns in encrypted data, thereby minimizing false negatives.

Figure 19 compares the assessed FPR using the proposed FHE-cPINN and existing approaches. The existing approaches for RF, NB, KNN, DT and SVM provide FPR values of 9.7%, 11.76%, 14.02%, 18.1%, and 14.57%, respectively. However, the FPR score of the proposed model is 1.98%. FHE-cPINN's low FPR demonstrates its ability to reduce false alarms, driven by physics-informed regularization and encrypted learning that better constrain prediction errors than traditional statistical models. Figure 20 compares the FNR metrics of the proposed and existing techniques. The FNR values of earlier classifiers, including RF, NB, KNN, DT and SVM, were 12.55%, 10.22%, 17.19%, 16.03%, and 9.77%, respectively. The proposed FHE-cPINN generates 3.95% of the FNR value. The reduced FNR confirms FHE-cPINN's effectiveness in capturing true attacks, enabled by enriched data representation and secure, encrypted training that boosts robustness against attack variants.

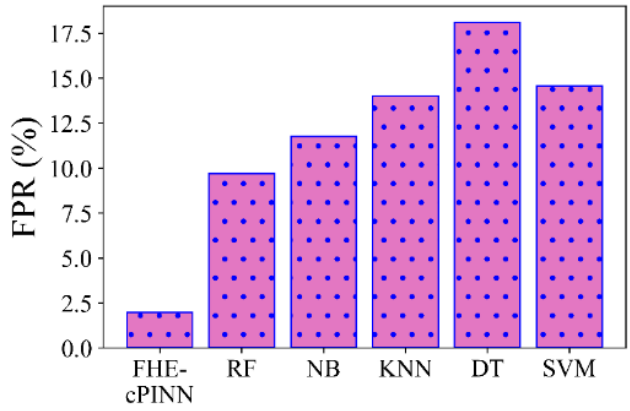

Figure 19. Comparison of False Positive Rate (FPR)

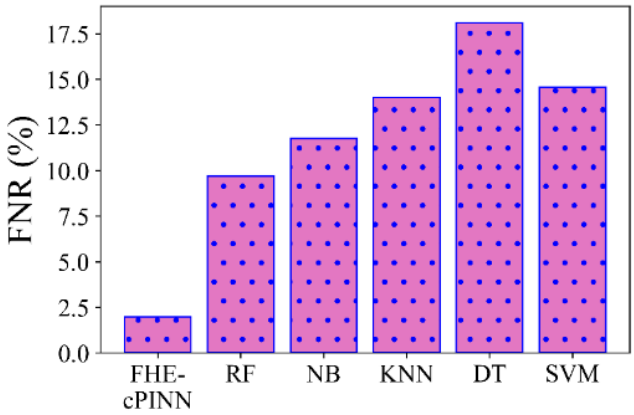

Figure 20. Comparison of False Negative Rate (FNR)

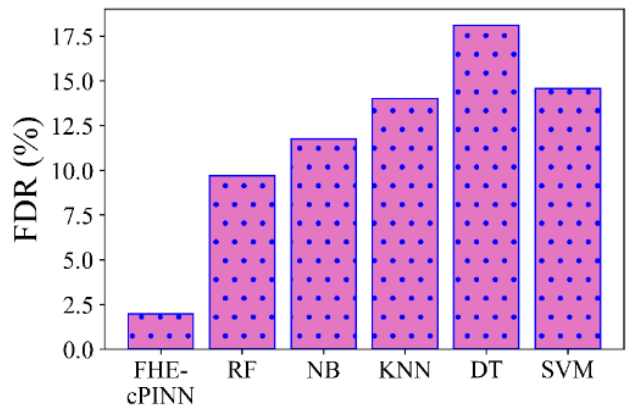

Figure 21. Comparison of False Discovery Rate (FDR)

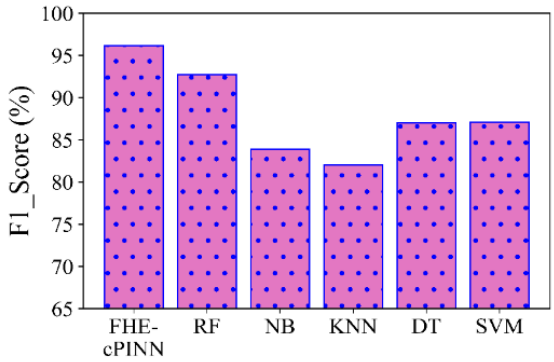

Figure 22. Comparison of F1-score

The FDR parameter for the proposed and existing approaches is shown in Figure 21. The proposed approach's FDR value of 3.71% was found to be higher than those of 11.68%, 13.13%, 14.55%, 15.5%, and 13.9% for existing techniques such as RF, NB, KNN, DT, and SVM. Lower FDR in FHE-cPINN reflects fewer incorrect positive identifications, aligning with enhanced precision and domain-guided learning that mitigates spurious correlations often seen in traditional classifiers. Figure 22 displays the existing and proposed F1_Score measurements. The proposed FHE-cPINN classifier achieves a score of 96.17%, which is higher than the F1 scores of 92.72%, 83.9%, 82%, 87%, and 87.1% reported by other existing systems, such as RF, NB, KNN, DT, and SVM. The F1-score shows FHE-cPINN's strong balance between precision and recall, reflecting its ability to detect attacks accurately while minimizing false alarms.

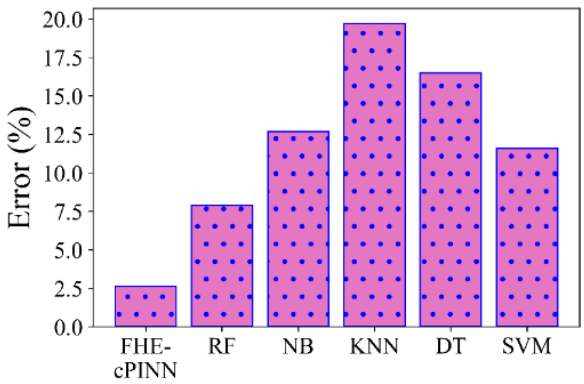

Figure 23. Comparison of error

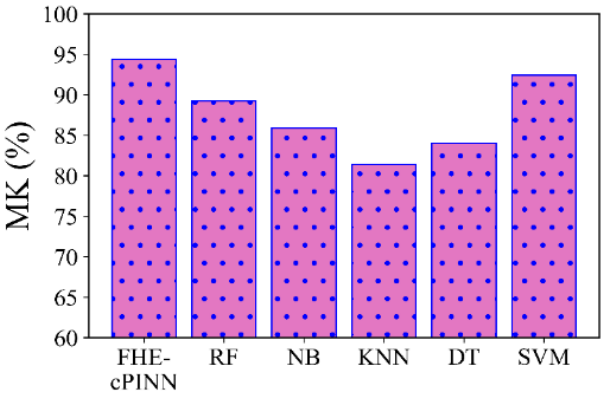

Figure 24. Comparison of MK

Figure 23 presents an analysis evaluating the proposed and existing error values. RF, NB, KNN, DT, and SVM have error rates of 7.88%, 12.7%, 19.7%, 16.5%, and 11.6%, respectively; in contrast, the proposed classifier has an error rate of 2.63%. This study assesses how the proposed classifier performs compared to alternative methods. FHE-cPINN's lower overall error rate indicates enhanced generalization and robustness, achieved through encrypted training and domain knowledge that reduce overfitting. A few conventional methods and the markedness value of the proposed model are contrasted in Figure 24. A system's ability to reliably forecast both positive and negative outcomes is measured by its markedness (MK). The comparison reveals that RF has 89.22% MK, NB has 85.9%, KNN has 81.4%, DT has 84%, and SVM has 92.4% MK. The proposed approach, FHE-cPINN, achieves 94.36%, outperforming more conventional techniques, according to the results. Higher MK in FHE-cPINN signals stronger predictive reliability across both classes, driven by its improved classification under encryption constraints and physics-based regularization.

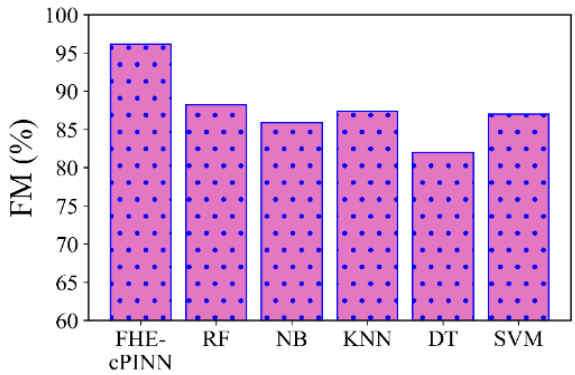

Figure 25. Comparison of F-Measure (FM)

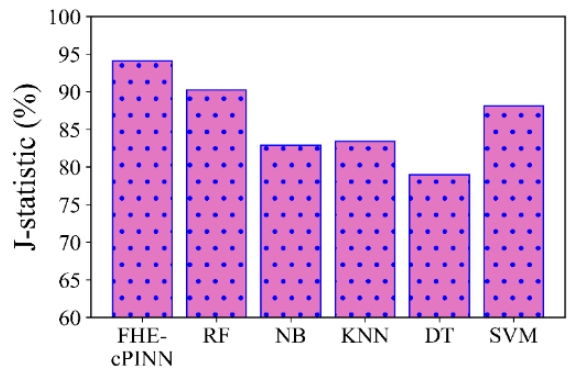

Figure 26. Comparison of J-statistic

The similarity between the two clusters is measured using the Fowlkes-Mallows index (FM). The FM comparison is displayed in Figure 25. Existing techniques such as RF, NB, KNN, DT, and SVM have FM values of 88.23%, 85.9%, 87.4%, 82%, and 87%, respectively; the proposed strategy, FHE-cPINN, achieves a 96.17% FM value. FHE-cPINN's higher FM indicates better separation of attack and non-attack classes, enabled by its integration of homomorphic encryption and physics-informed modeling. J-statistic of the FHE-cPINN and the existing classifiers are contrasted in Figure 26. FHE-cPINN yields a J-statistic value of 94.08%. For the RF, NB, KNN, DT and SVM classifiers, the existing J-statistic values

are 90.23%, 82.9%, 83.4%, 79%, and 88.1%, respectively. This shows that the proposed framework performs better than the existing, acknowledged methods. FHE-cPINN's higher value indicates stronger overall discrimination, driven by enriched training within encrypted, physics-informed frameworks.

4.3.1 Comparative study for state-of-the-art methods with the proposed model

The proposed model's performance is compared with several alternative models under shading conditions, as shown in Table 3. The metrics are considered as accuracy (%), specificity (%), precision (%), and F1-Score (%). Cleary shows that homomorphic encryption, named as CKKS-based cPINN (FHE-cPINN) model, not only offers high accuracy but also a fast execution time, which is 10 times faster than the Modified Bat Algorithm-one-class support vector machine (MBA-OCSVM) [24], deep convolutional neural network (DCNN) [25], Multi-Agent Reinforcement Learning (MARL) [26], and Long Short-Term Memory (LSTM) [27].

Table 3 analysis reveals that, while the error rates are lower for the proposed strategy than for the existing methods, the predictive values are significantly higher. Existing classifiers are inadequate for unsupervised data classification and fail when applied to independent predictors. So, homomorphic encryption, named as CKKS, based on the cPINN algorithm, was employed in this paper to improve autonomous vehicles' (CAVs') information security by identifying potential malicious messages.

Table 3. Comparative study of state-of-the-art methods

Methods	Accuracy	FNR	Precision	F1-Score
Proposed FHE-cPINN	97.37%	3.95%	96.29%	96.17%
MBA-OCSVM	95.50%	9.42%	94.22%	92.31%
DCNN	96.58%	13.55%	93.64%	93.45%
MARL	82.7%	18.79%	84.87%	81.43%
LSTM	94.2%	22.21%	91.86%	90.72%

4.4 Ablation study for the proposed model

Ablation experiments are essential for understanding the way various hyperparameters influence a neural network's learning process and overall performance. They provide information on the fine-tuning required for optimal model performance. The optimal hyperparameters for the proposed cPINN include a learning rate of 0.01, an input layer activation function of ReLU, and an optimizer of Adam.

Table 4 presents the ablation study results for varying learning rate, activation function, and optimizer when predicting intrusions on the CAN bus. Efficiency for evaluation by varying the learning rate of cPINN from 0.1, 0.01, and 0.001, activation function is from ReLU, tanh, and sigmoid, and optimizer is from Adam, SGD and NADAM. Among these, the proposed approach with a learning rate of 0.01, ReLU activation function, and Adam optimizer achieves better performance.

Table 4. Ablation study results for various parameters

Metrics	Learning Rate			Activation Function			Optimizer		
	0.1	0.01	0.001	ReLu	tanh	sigmoid	Adam	SGD	NADAM
Accuracy	91.25%	97.37%	95.27%	97.37%	95.24%	96.32%	97.37%	96.52%	92.67%
Recall	89.23%	96.29%	94.56%	96.29%	90.31%	94.73%	96.29%	92.32%	94.37%
F-Measure (FM)	89.17%	96.17%	93.25%	96.17%	91.52%	93.82%	96.17%	91.22%	94.77%
Net Present Value (NPV)	94.23%	98.07%	95.87%	98.07%	93.27%	94.23%	98.07%	94.17%	93.29%

Table 5. Cross-validation of the proposed models

Runs	Accuracy	Precision	F1-Score	Selectivity
1	97.11%	95.88%	95.74%	98.11%
2	97.42%	96.17%	96.22%	98.44%
3	97.84%	96.82%	96.66%	97.85%
4	97.09%	96.01%	95.95%	97.66%
5	97.39%	96.56%	96.29%	98.03%
Mean	97.37%	96.29%	96.17%	98.02%
Std Dev	±0.31	±0.42	±0.35	±0.28

Table 5 presents the results of a 5-run validation evaluating a classification model using four key metrics: accuracy, precision, recall, and selectivity. The mean values accuracy (97.37%), precision (96.29%), F1-score (96.17%), and selectivity (98.02%) indicate that the model is highly effective at correctly identifying both positive and negative cases, with a strong balance between detecting true positives and avoiding false positives. The low standard deviations for each metric confirm that the model's performance is stable and reliable across all folds.

Real-world scenario

An autonomous taxi operating in a smart city suddenly accelerates without driver input when a hacker injects fake messages into its CAN bus. The vehicle's safety could be seriously compromised, along with the safety of passengers and pedestrians. To prevent this, the car is equipped with a homomorphic encryption-based IDS that processes CAN bus data using BN and imputes missing values using random drop. For example, if a normal engine RPM signal is suddenly replaced with a spiked value, the system identifies this as abnormal. PHATE reduces data complexity, and the encrypted data is analyzed in the cloud using a CKKS-based cPINN, which detects anomalies while keeping the data encrypted for privacy. With fast detection, the system immediately alerts the vehicle to shift into safe mode, preventing a potential accident while ensuring cybersecurity and passenger safety.

5. CONCLUSION

The growing integration of ECUs and reliance on CAN bus communication in modern vehicles has significantly advanced automotive functionality but also introduced serious cybersecurity vulnerabilities. Traditional IDS, while useful, struggle with limitations in computational efficiency and detection accuracy, especially for complex, non-periodic, or subtle attacks like drop assaults. To address those limitations, a homomorphic Encryption-based machine learning model

was developed to secure the vehicle's CAN bus and protect it from malicious attacks. Datasets collected from real CAN traffic data pre-processed using Random drop imputation and BN. The dataset's dimensionality was reduced using PHATE. Finally, message attacks in CAN were identified, and the vehicle network was secured against cyber threats using homomorphic encryption, namely the CKKS-based cPINN. The modified TrustNet technique was used to defend against and mitigate attacks on CAN bus-critical communications, with minimal maintenance. Additionally, the effectiveness of the proposed FHE-cPINN classifier was compared with that of various techniques, including SVM, NB, KNN, DT and RF. For this proposed model, performance measures such as the F1-score, FM, MK, NPV, FDR, FPR, Accuracy, Precision, Error, and Specificity are evaluated. 96.17%, 96.17%, 98.07%, 3.71%, 1.98%, 97.37%, 96.29%, 2.63%, and 98.02% are the performance metrics achieved by the proposed approach. CKKS encryption was used to secure the system against cyberattacks; the encryption and decryption times were 106.57 seconds and 47.28 seconds, respectively. Compared with the existing model, the proposed model produces more accurate results.

Limitation and Future Scope

The lack of native FHE support in automotive microcontrollers can be mitigated by developing hardware-software co-design solutions tailored to embedded environments. The CKKS-based cPINN model faces challenges such as high latency, overfitting, error propagation in encrypted data, complex key management, loss of subtle signals due to dimensionality reduction, limited protocol support, and increased system complexity. Future improvements include model optimization, efficient encryption with hardware acceleration, better generalization through diverse data, error-resilient encryption, automated key management, adaptive feature selection, multi-protocol support, and modular, secure system design for easier maintenance.

REFERENCES

[1] Kanger, L., Geels, F.W., Sovacool, B., Schot, J. (2019). Technological diffusion as a process of societal embedding: Lessons from historical automobile transitions for future electric mobility. Transportation Research Part D: Transport and Environment, 71: 47-66. https://doi.org/10.1016/j.trd.2018.11.012

[2] Shen, Y., Cui, J., Zhong, H., Zhang, J., Bolodurina, I., He, D. (2024). A two-layer dynamic ECU group management scheme for in-vehicle CAN bus. IEEE Transactions on Intelligent Transportation Systems, 25(8): 10431-10445. https://doi.org/10.1109/TITS.2024.3384241

[3] Elkhail, A.A., Refat, R.U.D., Habre, R., Hafeez, A., Bacha, A., Malik, H. (2021). Vehicle security: A survey of security issues and vulnerabilities, malware attacks and defenses. IEEE Access, 9: 162401-162437. https://doi.org/10.1109/ACCESS.2021.3130495

[4] Humayed, A., Li, F., Lin, J., Luo, B. (2020). Cansentry: Securing can-based cyber-physical systems against denial and spoofing attacks. European Symposium on Research in Computer Security, 12308: 153-173. https://doi.org/10.1007/978-3-030-58951-6_8

[5] Karthick, S. (2018). TDP: A novel secure and energy aware routing protocol for Wireless Sensor Networks. International Journal of Intelligent Engineering and Systems, 11(2): 76-84. https://doi.org/10.22266/ijies2018.0430.09

[6] Martínez-Cruz, A., Ramírez-Gutiérrez, K.A., Feregrino-Uribe, C., Morales-Reyes, A. (2021). Security on in-vehicle communication protocols: Issues, challenges, and future research directions. Computer Communications, 180: 1-20. https://doi.org/10.1016/j.comcom.2021.08.027

[7] El-Rewini, Z., Sadatsharan, K., Selvaraj, D.F., Plathottam, S.J., Ranganathan, P. (2020). Cybersecurity challenges in vehicular communications. Vehicular Communications, 23: 100214. https://doi.org/10.1016/j.vehcom.2019.100214

[8] Hossain, M.D., Inoue, H., Ochiai, H., Fall, D., Kadobayashi, Y. (2020). LSTM-based intrusion detection system for in-vehicle CAN bus communications. IEEE Access, 8: 185489-185502. https://doi.org/10.1109/ACCESS.2020.3029307

[9] He, Q., Meng, X., Qu, R., Xi, R. (2020). Machine learning-based detection for cyber security attacks on connected and autonomous vehicles. Mathematics, 8(8): 1311. https://doi.org/10.3390/math8081311

[10] Yaacoub, J.P.A., Salman, O., Noura, H.N., Kaaniche, N., Chehab, A., Malli, M. (2020). Cyber-physical systems security: Limitations, issues and future trends. Microprocessors and Microsystems, 77: 103201. https://doi.org/10.1016/j.micpro.2020.103201

[11] Altalbe, A. (2023). Enhanced intrusion detection in in-vehicle networks using advanced feature fusion and stacking-enriched learning. IEEE Access, 12: 2045-2056. https://doi.org/10.1109/ACCESS.2023.3347619

[12] Olufowobi, H., Young, C., Zambreno, J., Bloom, G. (2019). SAIDuCANT: Specification-based automotive intrusion detection using controller area network (CAN) timing. IEEE Transactions on Vehicular Technology, 69(2): 1484-1494. https://doi.org/10.1109/TVT.2019.2961344

[13] Barletta, V.S., Caivano, D., Nannavecchia, A., Scalera, M. (2020). Intrusion detection for in-vehicle communication networks: An unsupervised Kohonen SOM approach. Future Internet, 12(7): 119. https://doi.org/10.3390/fi12070119

[14] Pascale, F., Adinolfi, E.A., Coppola, S., Santonicola, E. (2021). Cybersecurity in automotive: An intrusion detection system in connected vehicles. Electronics, 10(15): 1765. https://doi.org/10.3390/electronics10151765

[15] Song, H.M., Woo, J., Kim, H.K. (2020). In-vehicle network intrusion detection using deep convolutional neural network. Vehicular Communications, 21: 100198. https://doi.org/10.1016/j.vehcom.2019.100198

[16] Jeong, S., Jeon, B., Chung, B., Kim, H.K. (2021). Convolutional neural network-based intrusion detection system for AVTP streams in automotive Ethernet-based networks. Vehicular Communications, 29: 100338. https://doi.org/10.1016/j.vehcom.2021.100338

[17] Choi, T.M., Kang, J.S., Kim, J.H. (2023). RDIS: Random drop imputation with self-training for incomplete time series data. IEEE Access, 11: 100720-100728. https://doi.org/10.1109/ACCESS.2023.3315343

[18] Bilal, M.A., Ji, Y., Wang, Y., Akhter, M.P., Yaqub, M. (2022). An early warning system for earthquake

prediction from seismic data using batch normalized graph convolutional neural network with attention mechanism (BNGCNNATT). Sensors, 22(17): 6482. https://doi.org/10.3390/s22176482

[19] Bao, W., Gu, Y., Chen, B., Yu, H. (2023). Golgi_DF: Golgi proteins classification with deep forest. Frontiers in Neuroscience, 17: 1197824. https://doi.org/10.3389/fnins.2023.1197824

[20] Cheon, J.H., Costache, A., Moreno, R.C., Dai, W., Gama, N., Georgieva, M., Song, Y. (2021). Introduction to homomorphic encryption and schemes. In Protecting Privacy through Homomorphic Encryption, pp. 3-28. https://doi.org/10.1007/978-3-030-77287-1_1

[21] Jagtap, A.D., Kharazmi, E., Karniadakis, G.E. (2020). Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. Computer Methods in Applied Mechanics and Engineering, 365: 113028. https://doi.org/10.1016/j.cma.2020.113028

[22] Kornaros, G., Bakoyiannis, D., Tomoutzoglou, O., Coppola, M., Gherardi, G. (2019). Trustnet: Ensuring normal-world and trusted-world can-bus networking. In 2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), Beijing, China, pp. 1-6. https://doi.org/10.1109/SmartGridComm.2019.8909715

[23] HCRL-CAN-Intrusion-Dataset, https://ocslab.hksecurity.net/Datasets/CAN-intrusion-dataset, accessed on 14-06-2024.

[24] Avatefipour, O., Al-Sumaiti, A.S., El-Sherbeeny, A.M., Awwad, E.M., Elmeligy, M.A., Mohamed, M.A., Malik, H. (2019). An intelligent secured framework for cyberattack detection in electric vehicles' CAN bus using machine learning. IEEE Access, 7: 127580-127592. https://doi.org/10.1109/ACCESS.2019.2937576

[25] Khandelwal, S., Wadhwa, E., Shreejith, S. (2022). Deep learning-based embedded intrusion detection system for automotive CAN. In 2022 IEEE 33rd International Conference on Application-Specific Systems, Architectures and Processors (ASAP), Gothenburg, Sweden, pp. 88-92. https://doi.org/10.1109/ASAP54787.2022.00023

[26] Prathiba, S.B., Raja, G., Anbalagan, S., Arikumar, K.S., Gurumoorthy, S., Dev, K. (2022). A hybrid deep sensor anomaly detection for autonomous vehicles in 6G-V2X environment. IEEE Transactions on Network Science and Engineering, 10(3): 1246-1255. https://doi.org/10.1109/TNSE.2022.3188304

[27] Alsulami, A.A., Abu Al-Haija, Q., Alqahtani, A., Alsini, R. (2022). Symmetrical simulation scheme for anomaly detection in autonomous vehicles based on LSTM model. Symmetry, 14(7): 1450. https://doi.org/10.3390/sym14071450